

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-02-3-ErreurstypeBool-pt5

Concepts (extraits des sous-titres générés automatiquement) :

Réponse a.. Variable de type booléen. Type boolean. Branchement conditionnel. Instruction suivante. Égal b. Déclaration de variable booléenne. Type des conditions. Réponse. Opérateurs logiques. Condition. Fois n. Intérieur d'un branchement conditionnel. Variables. Type int.



[vers la recherche de séquences vidéo](#)
(dans Initiation à la programmation (en Java).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

Erreurs de débutant

Le type `boolean` (Partie 5)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s



```

System.out.println("Entrez le premier nombre:");
int n = scanner.nextInt();
System.out.println("Entrez le deuxieme nombre:");
int p = scanner.nextInt();
if ((n < p) && (2 * n >= p)) {
    System.out.print("1");
}
if ((n < p) || (2 * n >= p)) {
    System.out.print("2");
}
if (n < p) {
    if (2 * n >= p) {
        System.out.print("3");
    } else {
        System.out.print("4");
    }
}
System.out.println();

```



- A: 2 2
 B: 24
 C: 123
 D: 1234

Qu'affiche ce programme quand l'utilisateur entre 2 et 1 ?

réponse est la réponse A. Pour ce dernier exercice, n a la valeur 2 et p a la valeur 1. Cette condition est donc cette fois-ci fausse puisque 2 n'est pas strictement inférieur à 1. Comme il s'agit d'un ET ici, je n'ai pas besoin d'évaluer cette condition-ci, je sais déjà que toute cette condition est forcément fausse. Je passe donc à ce branchement conditionnel-ci. Cette condition, et bien c'est la même qu'avant, j'ai dit qu'elle était fausse, mais il s'agit d'un OU ici, donc je suis obligé d'évaluer cette expression-ci, cette condition-ci, 2 fois n vaut 4, p vaut 1, comme 4 est supérieur ou égal à 1, et bien cette condition-ci est vraie. Comme il s'agit d'un OU, cette condition est vraie, on va donc exécuter cette instruction, c'est-à-dire afficher un 2. On passe ensuite ici. Cette condition est fausse, c'est encore une fois la même qu'avant, on saute donc ce branchement conditionnel pour arriver à ce stade du programme qui fait simplement un retour à la ligne.

notes

résumé

0m 1s



On peut initialiser des booléens à l'aide des constantes `false` et `true`.

On peut utiliser des booléens comme des conditions. Par exemple:

- on peut utiliser des opérateurs logiques (`&&`, `||` et `!`) entre booléens;
- on peut utiliser un booléen comme condition dans un `if`.

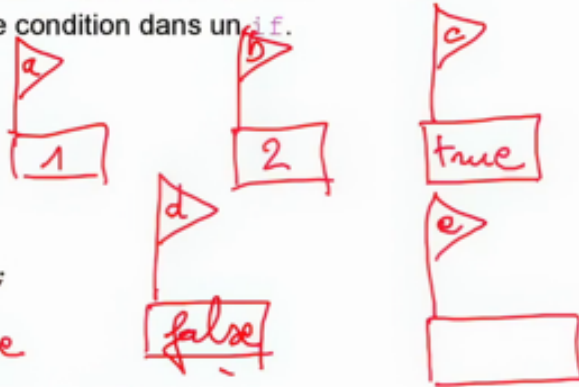
```
→ int a = 1, b = 2;
```

```
→ boolean c = true;
```

```
→ boolean d = (a == b);
```

```
→ boolean e = (d || (a < b));
```

```
if (e) {
    System.out.println("e vaut true");
}
```



Passons maintenant au type boolean ou booléen en français, qui est le type des conditions. Cet type va nous permettre de déclarer des variables qui vont contenir la valeur de condition. Une variable de type booléen est souvent appelée tout simplement un booléen et ne peut contenir que deux valeurs possibles, soit vrai, soit faux, ou plus exactement en Java, on va utiliser les valeurs littérales `true` et `false`. Dans cet exemple je commence par déclarer deux variables `a` et `b` de type `int`, `a` est initialisé à 1, `b` est initialisé à 2. Cette instruction est une déclaration de variable booléenne, cette variable s'appelle `test1` qui sera de type effectivement booléen. Comme toute variable, je peux la représenter par une boîte. Cette variable est initialisée à la valeur de la condition `a égal b`, `a` vaut 1, `b` vaut 2 donc cette condition est fausse et `test1` va être initialisé à faux ou plus exactement `false` qui veut dire faux en anglais. Cette instruction suivante déclare une deuxième variable de type booléen qui s'appelle cette fois-ci `test2` qui va être initialisée à la valeur de la condition, `a strictement inférieur à b`. Comme `a` vaut 1 et `b` vaut 2, cette condition est vraie. Donc `test2` va être initialisé à vrai ou plus exactement à `true` qui veut dire vrai en anglais. Certains étudiants n'ont pas de problèmes avec les booléens, d'autres font parfois un blocage. Alors rappelez-vous qu'une variable de type boolean est une variable comme les autres, c'est-à-dire que vous pouvez vous la représenter comme une boîte. Il se trouve simplement que cette boîte ne peut contenir que deux valeurs possibles, soit `true` soit `false`. Il se trouve aussi qu'on peut utiliser des opérateurs logiques entre les variables de type bool et qu'on peut utiliser ces variables comme des conditions, par exemple à l'intérieur d'un branchement conditionnel et c'est ce que nous allons voir sur cet exemple.

notes

résumé

1m 37s



On peut initialiser des booléens à l'aide des constantes `false` et `true`.

On peut utiliser des booléens comme des conditions. Par exemple:

- on peut utiliser des opérateurs logiques (`&&`, `||` et `!`) entre booléens;
- on peut utiliser un booléen comme condition dans un `if`.

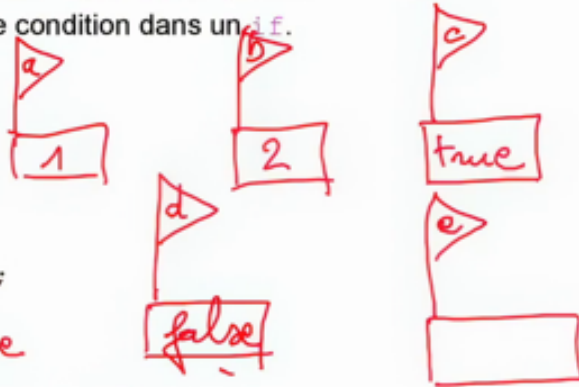
→ `int a = 1, b = 2;`

→ `boolean c = true;`

→ `boolean d = (a == b);`

→ `boolean e = (d || (a < b));`

```
if (e) {
    System.out.println("e vaut true");
}
```



Comme dans l'exemple précédent, je commence ici par déclarer une variable `a` de type `int` initialisée à 1, une variable `b` de type `int` également, initialisée à 2. Dans l'instruction suivante, je déclare cette fois-ci une variable de type booléen qui s'appelle `c` et qui est initialisée à la valeur `true`. Dans l'instruction suivante je déclare un booléen qui s'appelle `d`, et qui est initialisée à la valeur de la condition `a égal b`. `a` vaut 1, `b` vaut 2, la condition est fausse, elle vaut donc `false`. `d` est donc initialisé à `false`. Dans l'instruction suivante, je déclare une variable de type booléen qui s'appelle `e` et qui est initialisée à la valeur de cette condition. `d` vaut `false`. Cette condition, `a inférieur à b` est vraie, elle vaut donc `true`. Comme l'opérateur

notes

résumé

logique ici est un OU, toute cette condition est vraie et elle vaut true. Donc e est initialisée à true. On passe maintenant à ce branchement conditionnel-ci qui a pour condition la valeur de la variable booléenne qui s'appelle e, qui vaut true, donc cette condition est vraie, et on va donc rentrer dans le branchement conditionnel pour exécuter cette instruction et donc afficher le message "e vaut true". Voilà, c'est tout pour le moment pour les booléens mais sachez qu'ils sont utiles pour beaucoup de problèmes et nous les rencontrerons dans de nombreux exemples. exemples.

[illegible]